Term_Project - CH5350

Shritej Chavan (BE14B004)

December 3, 2018

Threshold Auto-Regressive Model

a) A two regime TAR (2, d=1) model, has the form:

$$v[k] = -d_0^{(1)} - d_1^{(1)} v[k-1] - d_2^{(1)} v[k-2] + e_1[k], v[k-1] \leq \gamma, e_1[k] \sim i.i.d.(0, \sigma_1^2)$$

$$v[k] = -d_0^{(2)} - d_1^{(2)} v[k-1] - d_2^{(2)} v[k-2] + e_2[k], v[k-1] > \gamma, e_2[k] \sim i.i.d.(0, \sigma_2^2)$$

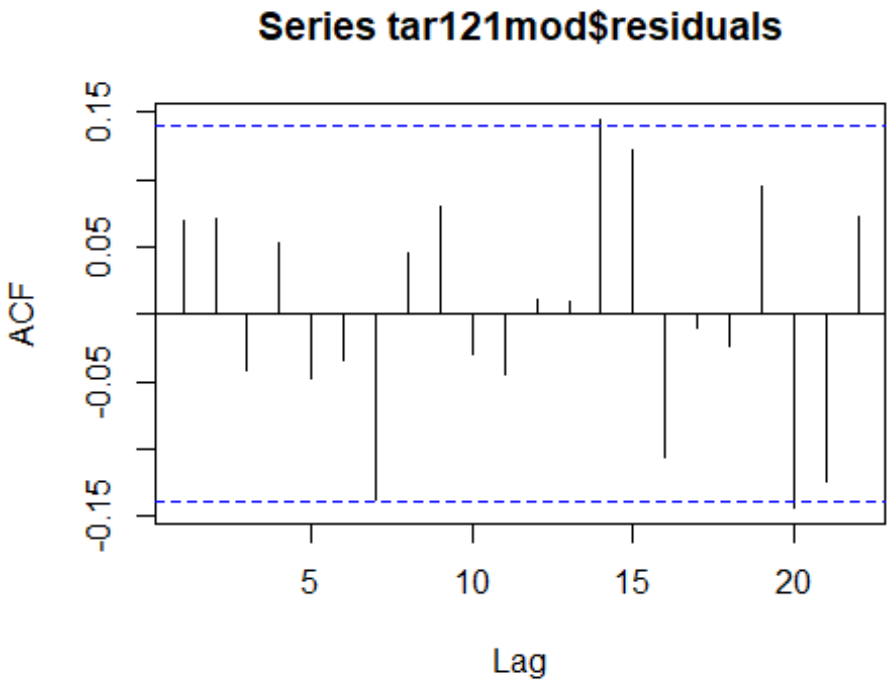
```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

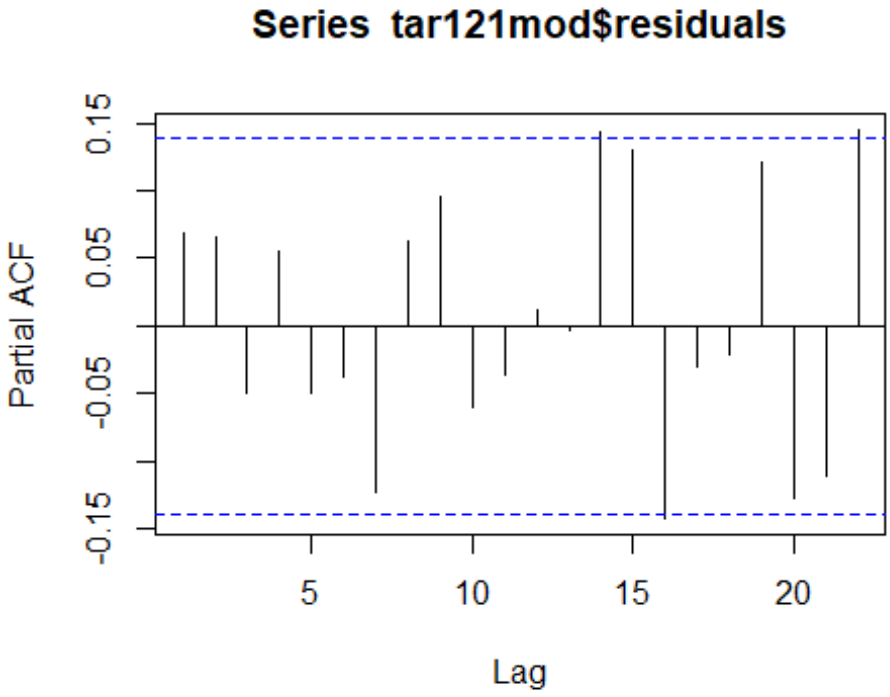
```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

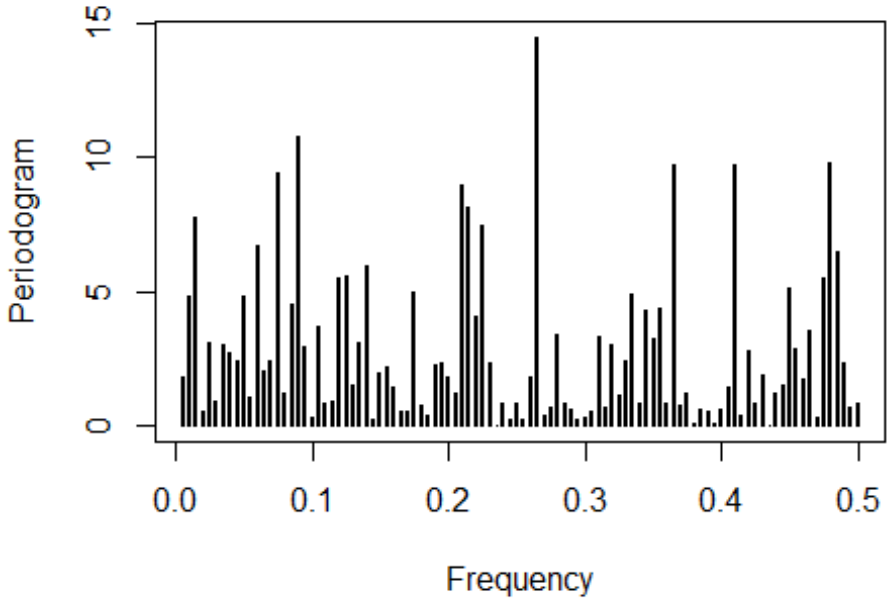
```
load("projq1a.Rdata")

tar121mod = tar(vk,1,2,1)

acf(tar121mod$residuals)
```




```
pacf(tar121mod$residuals)
```




```
periodogram(tar121mod$residuals)
```

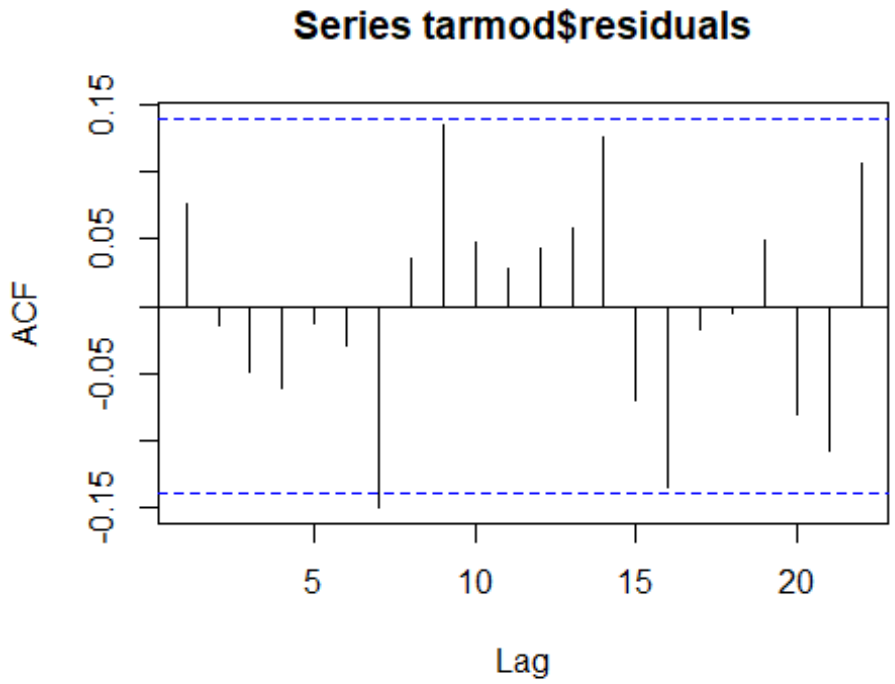


TAR(1,2,1) model satisfies all residuals tests.

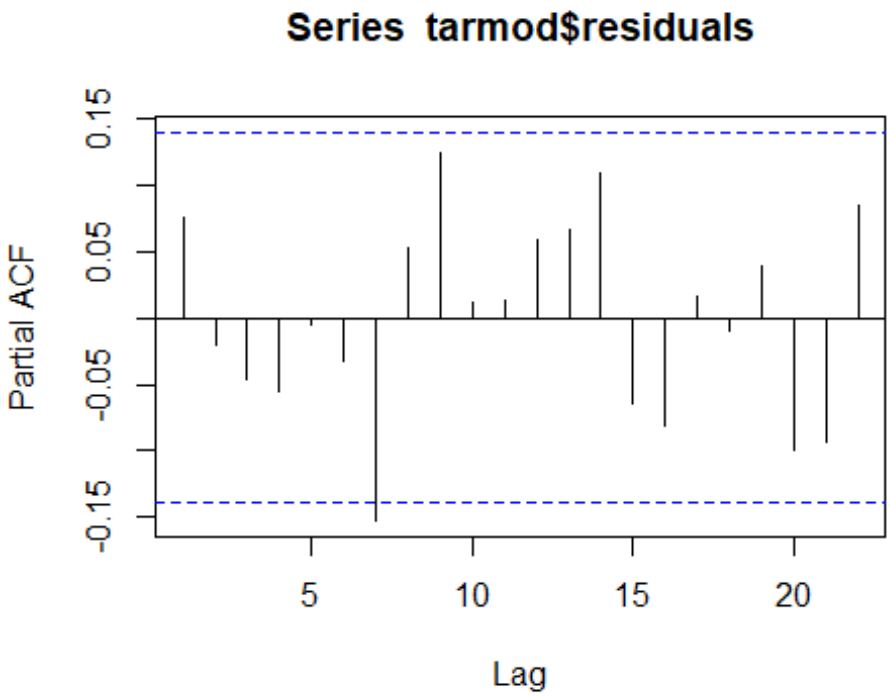
But let's check model with lower number of parameters to satisfy parsimony conditions

```
tarmod = tar(vk,2,0,2)

#Coefficients for the first regime of the fitted TAR(2,0,2) model
tarmod$qr1$coefficients

## intercept-vk      lag1-vk      lag2-vk
##   -0.7312821   -0.3731717    0.2474873

#Sigma of the residuals of the regimes
var(tarmod$qr1$residuals)   # Regime 1

## [1] 1.53172

var(tarmod$qr2$residuals)   # Regime 2

## [1] 0.9133471

#Threshold

tarmod$thd

##
## 0.877825
```

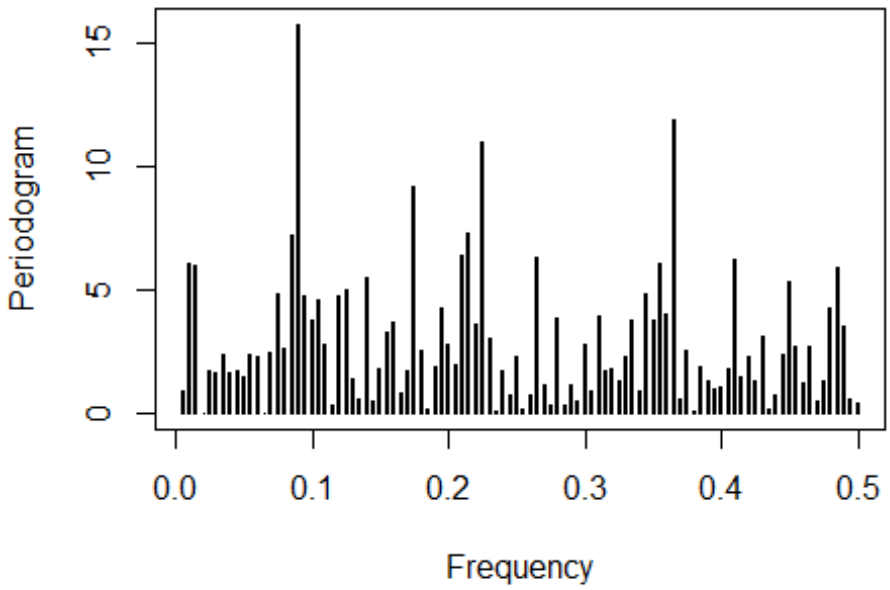
```r
#Checking the residuals of the TAR model


acf(tarmod$residuals)
```




```r
pacf(tarmod$residuals)
```




```r
periodogram(tarmod$residuals)
```



After viewing the ACF, PACF and periodogram of the residuals we can say that it passes the residual tests and the TAR(2,0,2) model perfectly fits on the given data set.

And the basis of parsimony TAR(2,0,2) model is chosen as the perfect fit for the given series.

The final two regime model can written as follows

$$v[k] = -0.73 - 0.37v[k-1] - 0.25v[k-2] + e_1[k], v[k-1] \leq 0.87, e_1[k] \sim i.i.d.(0,1.532)$$

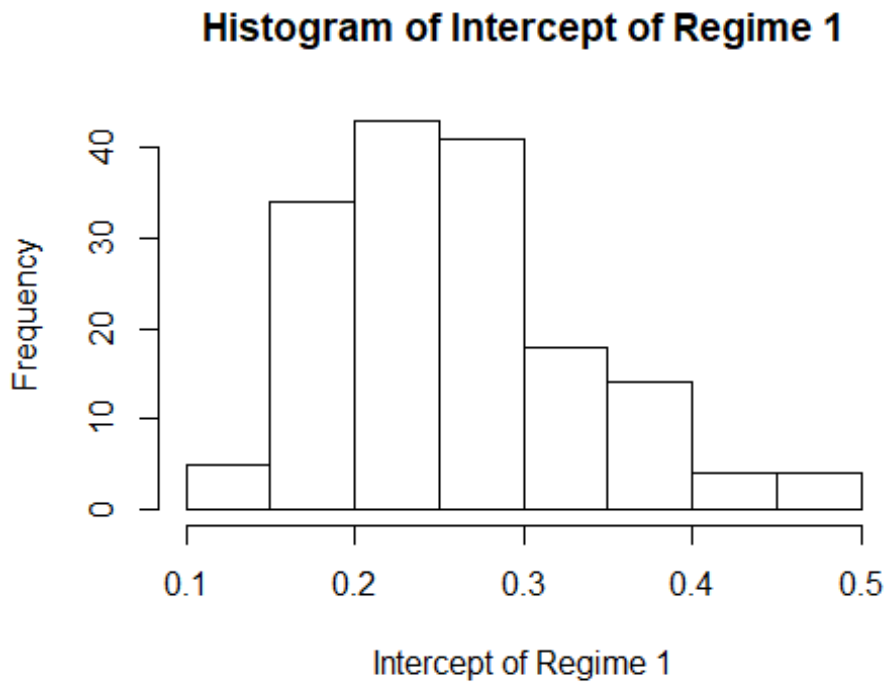
$$v[k] = 0.8 + e_2[k], v[k-1] > 0.87, e_2[k] \sim i.i.d.(0,0.913)$$

```
#Residual from the best model
epsk = tarmod$residuals

mod_coeff_reg1 = matrix(data = NA,  nrow = 200, ncol = 3)

mod_coeff_reg2 = {}

#Number of realizations
R = 1:200

for (i in R){

  #Resampling the residuals with replacement
  epskr1 <- sample(epsk, size= length(epsk), replace=T)

  # Generating new artificial realization of the series
  vk_new = tar.sim(tarmod, ntransient = 0, n = 198, epskr1)

  #Restimating model parameters using same orders and delays
  vmod = tar(vk_new$y[1:196],2,0,2)

  #Saving the model parameters
  mod_coeff_reg1[i,1] = vmod$qr1$coefficients[1]
  mod_coeff_reg1[i,2] = vmod$qr1$coefficients[2]
  mod_coeff_reg1[i,3] = vmod$qr1$coefficients[3]

  mod_coeff_reg2[i] = vmod$qr2$coefficients



}

#Omitting the NA values from the matrix of coefficients
#Index of the number
coef_q1_3 = which(is.na(mod_coeff_reg1[,3]) == F)
```

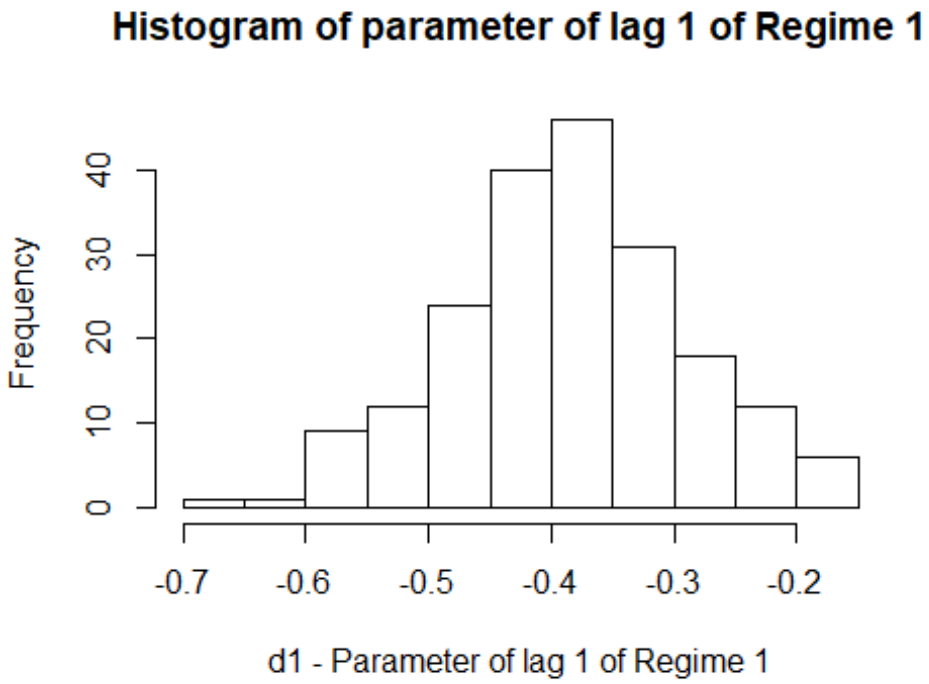
```
coef_q1_2 = which(is.na(mod_coeff_reg1[,2]) == F)
coef_q1_1 = which(is.na(mod_coeff_reg1[,1]) == F)
```

Now plotting the distribution (Histogram) of all the coefficients of the two regimes.

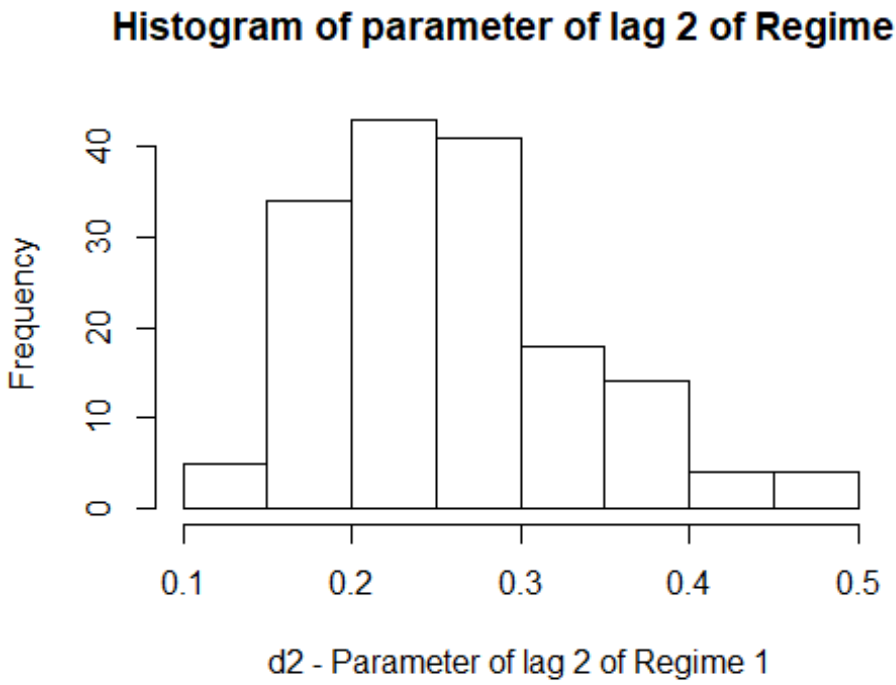
```
hist(mod_coeff_reg1[coef_q1_1,3], xlab = "Intercept of Regime 1", main =
"Histogram of Intercept of Regime 1")
```



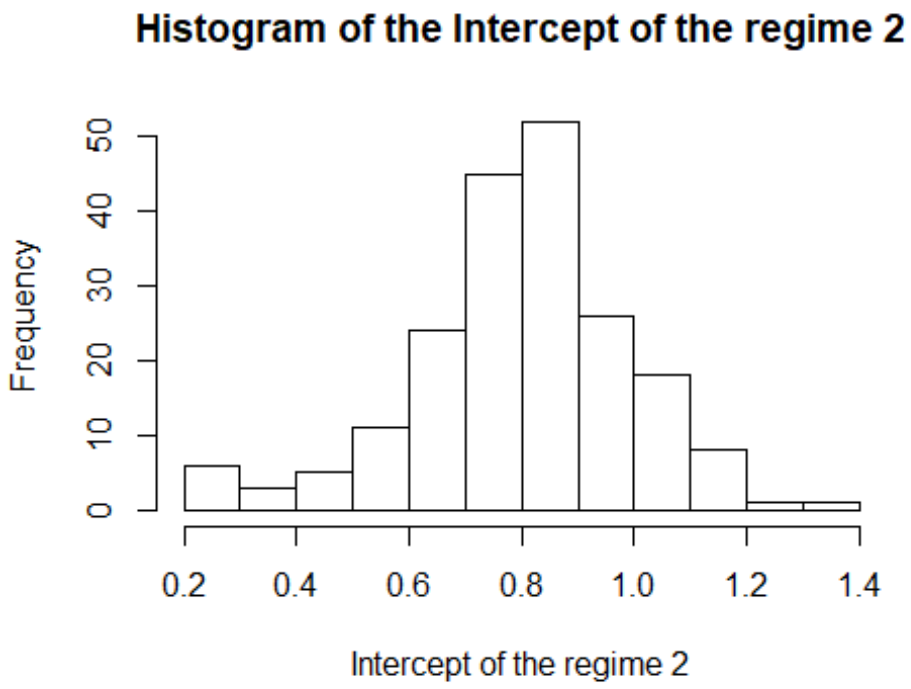

```
hist(mod_coeff_reg1[coef_q1_2,2], xlab = "d1 - Parameter of lag 1 of Regime
1", main = "Histogram of parameter of lag 1 of Regime 1")
```




```
hist(mod_coeff_reg1[coef_q1_3, 3], xlab = "d2 - Parameter of lag 2 of Regime
1", main = "Histogram of parameter of lag 2 of Regime 1")
```



```
hist(mod_coeff_reg2, xlab = "Intercept of the regime 2", main = "Histogram of
the Intercept of the regime 2")
```



Now calculating mean and variances of the distribution of the respective coefficients

```
#Mean of all the parameters in Regime 1
mean_reg1 = colMeans(mod_coeff_reg1,na.rm = TRUE)

#Mean of the parameters in Regime 2
mean_reg2 = mean(mod_coeff_reg2)

#Standard error of the parameters in Regime 2
std_reg2 = sqrt(var(mod_coeff_reg2))

#Standard error of the parameters in Regime 1
std_reg1 = matrix(data = c(sqrt(var(mod_coeff_reg1[coef_q1_1,1])),
sqrt(var(mod_coeff_reg1[coef_q1_2,2])),
sqrt(var(mod_coeff_reg1[coef_q1_3,3]))), nrow = 1, ncol = 3, byrow = TRUE )

means = c(mean_reg1, mean_reg2)
stds = c(std_reg1, std_reg2)

results = as.matrix(cbind(means, stds))

rownames(results)=c('Regime1_d0','Regime1_d1','Regime1_d2','Regime2_d0')
colnames(results)=c('Mean','Standard error')
```

```
results
```

```
##                   Mean Standard error
## Regime1_d0 -0.7731765     0.17388259
## Regime1_d1 -0.3862817     0.09382737
## Regime1_d2  0.2552201     0.07498505
## Regime2_d0  0.8005845     0.19681567
```

The value of the parameters calculated from the distribution lie in the 99% confidence interval (as shown by the mean and) of the true values which were obtained from the tar routine.

(S)ARIMA Model

```
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:TSA':
##
##     spec
```

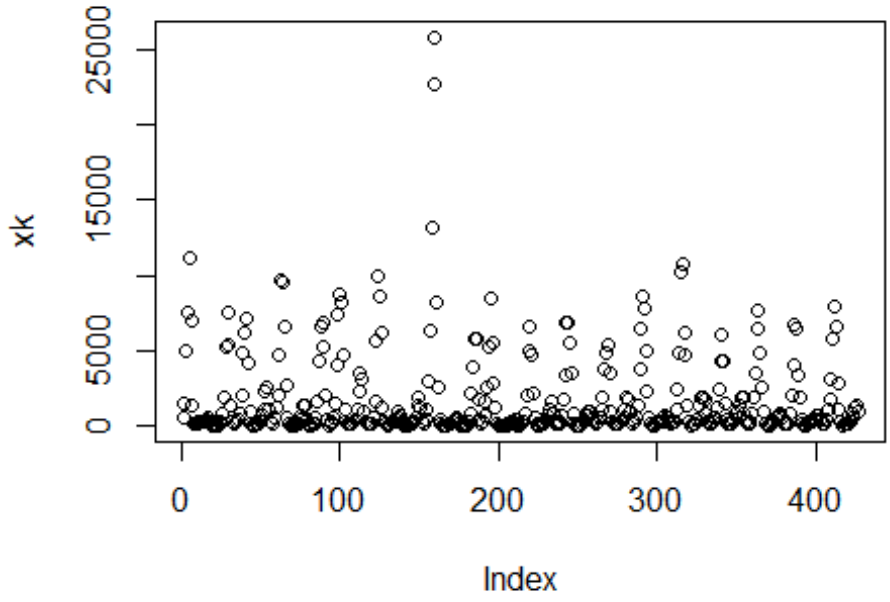
```
library(TSA)
```

```
monthly_reported_number_of_cases_1_ <- read_csv("monthly-reported-number-of-cases.csv")
```

```
## Parsed with column specification:
## cols(
##   Month = col_character(),
##   `Monthly reported number of cases of measles, New York city, 1928-1972` = col_character()
## )
```

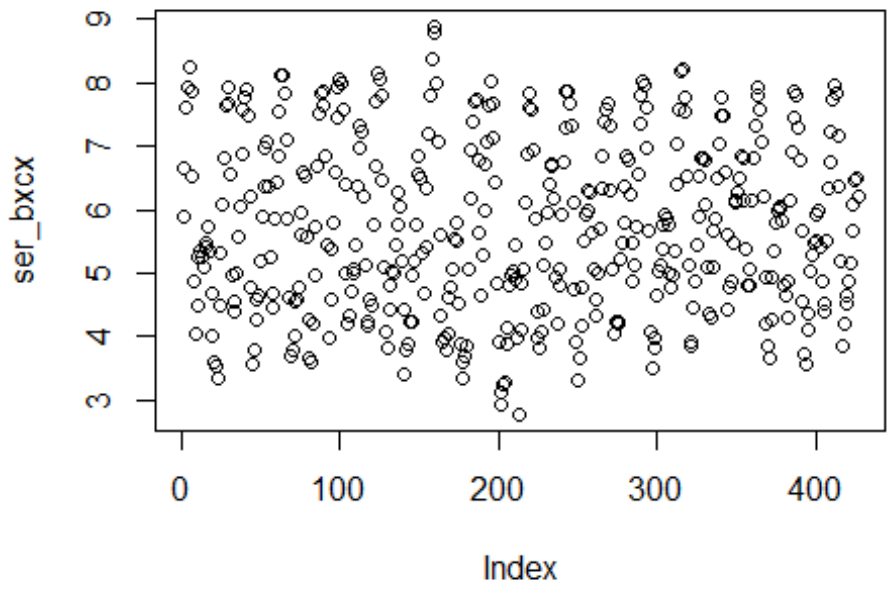
```
## Warning: 1 parsing failure.
## row col  expected    actual                                      file
## 535  -- 2 columns 3 columns 'monthly-reported-number-of-cases.csv'
```

```
xk_true = monthly_reported_number_of_cases_1_$`Monthly reported number of cases of measles, New York city, 1928-1972`

# Distribution of dataset, 80% - Training and 20% - Test data

xk = as.numeric(xk_true[1:427])

plot(xk)
```

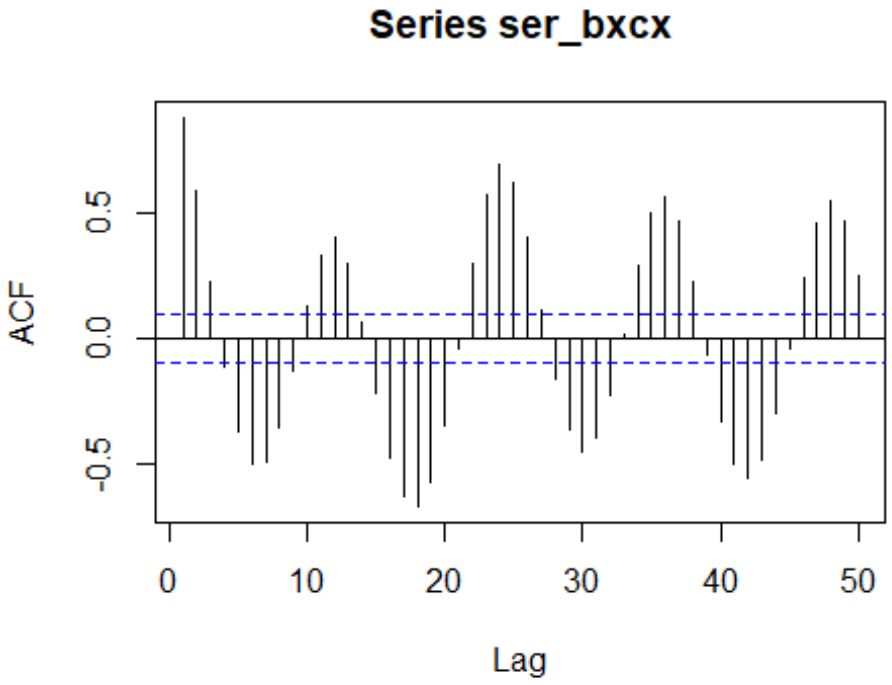


From the above plot of the series we can see that it is heteroskedastic i.e the variance is not constant throughout the series. Therefore to deal with this we will be using Box-Cox Transformation.

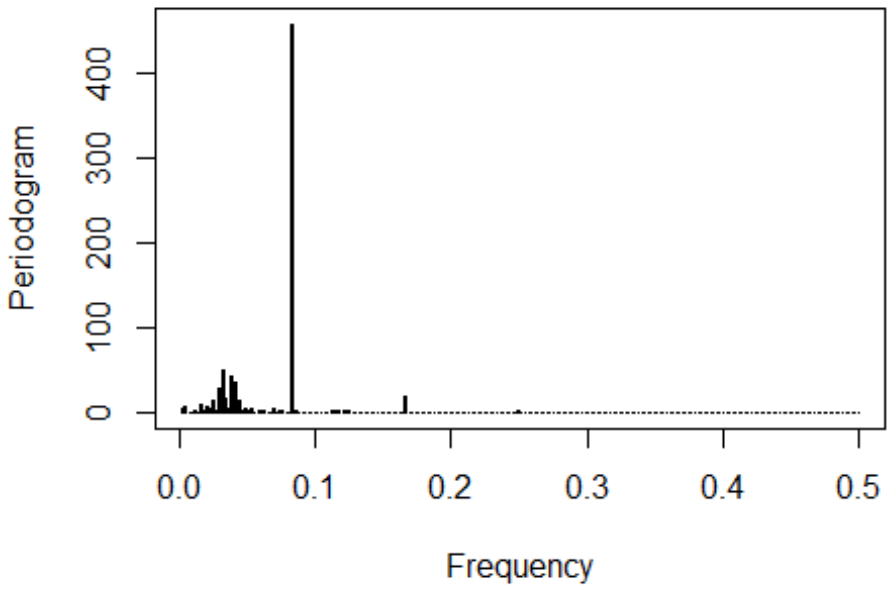
```
library(forecast)

lambda_bxcx = BoxCox.lambda(xk, method = 'guerrero', lower=-5,upper=5)
lambda_bxcx

## [1] -0.02713624

ser_bxcx = BoxCox(xk,lambda_bxcx)

plot(ser_bxcx)
```




```
acf(ser_bxcx, lag.max = 50)
```




```
periodogram(ser_bxcx)
```



Observing the periodogram we can say that the since there is no concentration of the psd at low frequency, there is no need for differencing.

Also, performing ADF (Augmented Dickey Fuller) test on the given series as follows

```
library(tseries)

adf.test(ser_bxcx)

## Warning in adf.test(ser_bxcx): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  ser_bxcx
## Dickey-Fuller = -5.9884, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

Null hypothesis is rejected , hence there isn't any need for differencing

Now observing the ACF plot of the series we can say that the series is periodic with period of 12.

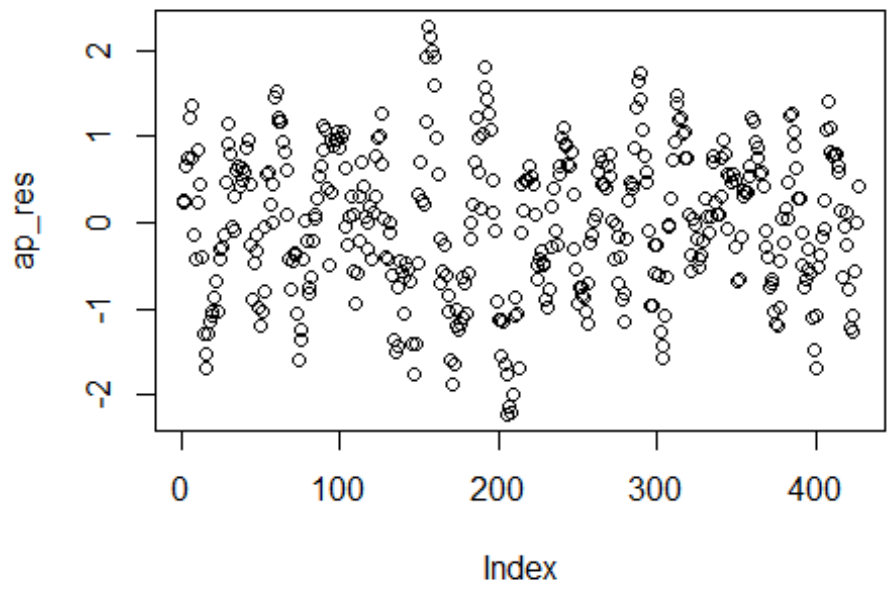
So there is a seasonal component but 'stl' function fails to detect seasonality.

Therefore we use 'lm' method to fit the seasonal component through sinusiodal functions.

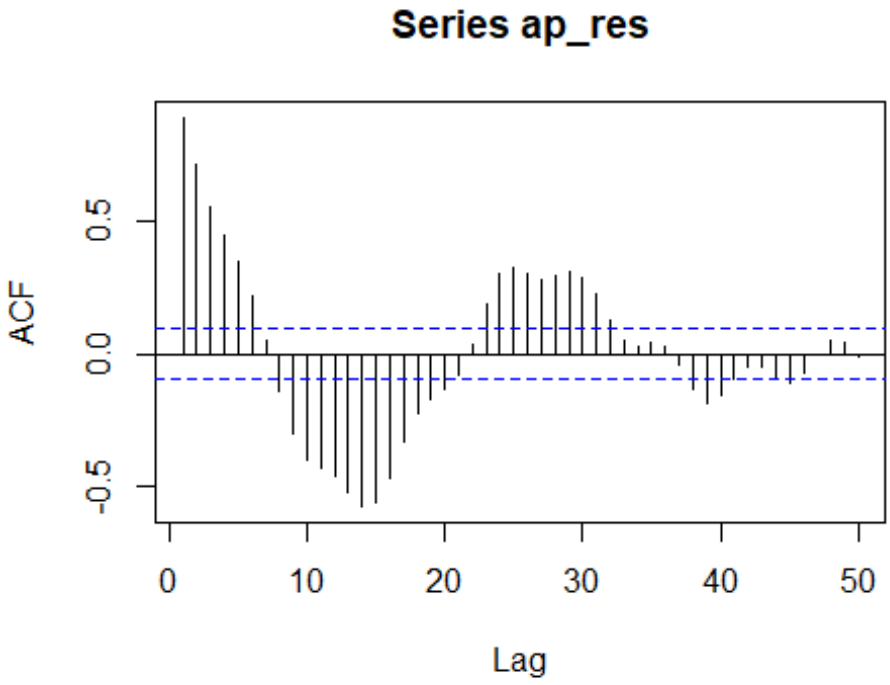
```r
lin = 1:length(ser_bxcx)

periodic = lm(ser_bxcx ~sin(2*pi*lin/12)+I(cos(2*pi*lin/12)))

summary(periodic)
```

```
##
## Call:
## lm(formula = ser_bxcx ~ sin(2 * pi * lin/12) + I(cos(2 * pi *
##     lin/12)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2268 -0.6148  0.0344  0.6538  2.2810
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.70829    0.04123  138.44   <2e-16 ***
## sin(2 * pi * lin/12)      1.23435    0.05834   21.16   <2e-16 ***
## I(cos(2 * pi * lin/12)) -0.79405    0.05827  -13.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8519 on 424 degrees of freedom
## Multiple R-squared:  0.5985, Adjusted R-squared:  0.5966
## F-statistic:   316 on 2 and 424 DF,  p-value: < 2.2e-16
```

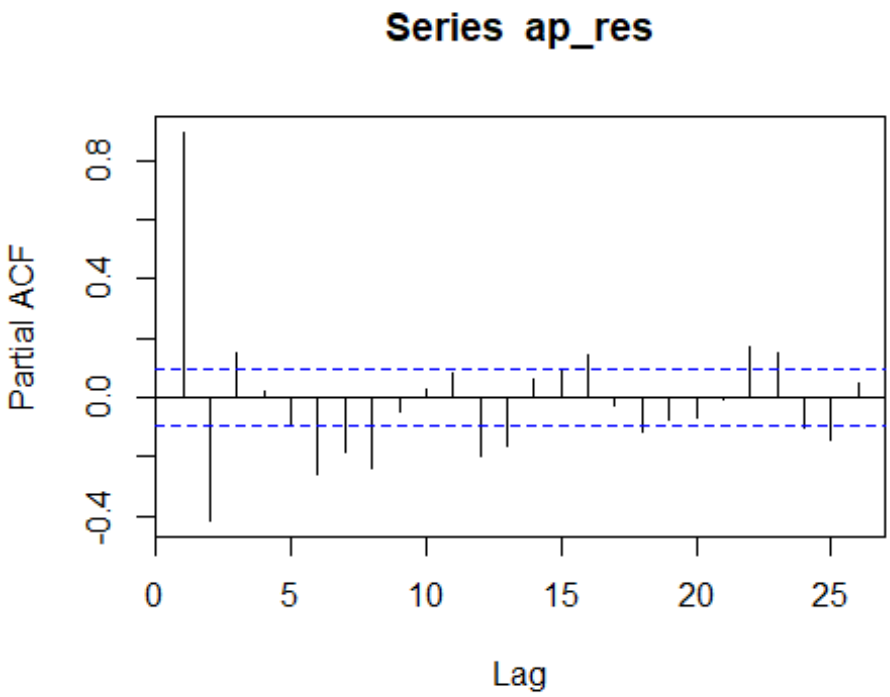
```r
ap_res = periodic$residuals

plot(ap_res)
```



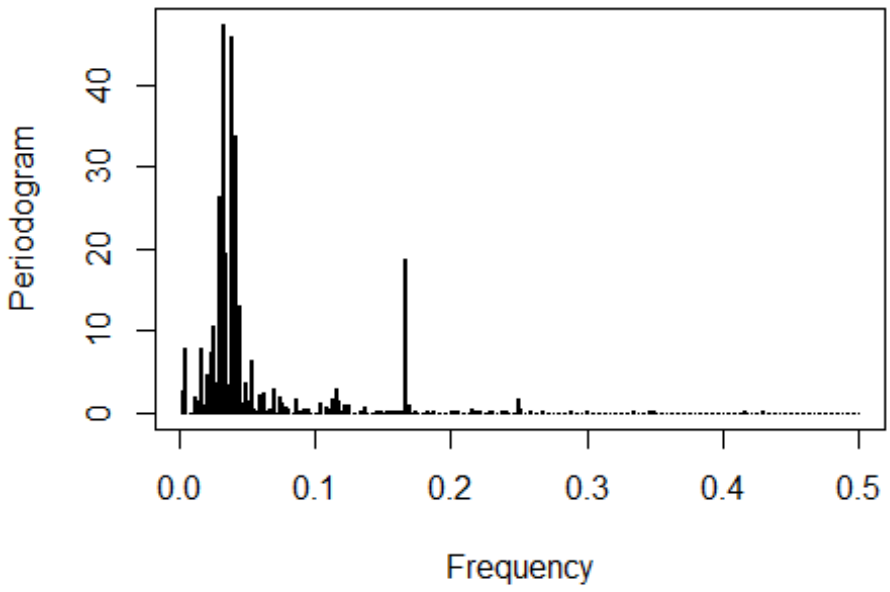

```
acf(ap_res, lag.max = 50)
```




```
pacf(ap_res)
```




```
periodogram(ap_res)
```



Periodogram of the residual series clearly shows that there is no integrating effects present in the series but seasonal component is, as seen from the ACF plot.

Therefore, fitting multiplicative SARIMA model by trial and error by restricting the differencing term to be zero on both list

```
sarima = arima(ap_res, order = c(0,0,1),c(1,0,2))

#tsdiag(sarima)

sarima

##
## Call:
## arima(x = ap_res, order = c(0, 0, 1), seasonal = c(1, 0, 2))
##
## Coefficients:
##          ma1    sar1    sma1    sma2  intercept
##       0.4312  0.7332  0.1740  0.1405     0.0076
## s.e.  0.2366  0.0707  0.1876  0.1234     0.1157
##
## sigma^2 estimated as 0.1167:  log likelihood = -148.21,  aic = 306.43
```

We can clearly see that the model clearly overfits the series.

Therefore we will fit the constrained SARIMA model.

```
sarima2=arima(ap_res,c(02,0,02),seasonal=list(order=c(0,0,2)) ,fixed =
c(0,NA,NA,NA,NA,0,NA),transform.pars = F)

sarima2

##
## Call:
## arima(x = ap_res, order = c(2, 0, 2), seasonal = list(order = c(0, 0, 2)),
transform.pars = F,
##     fixed = c(0, NA, NA, NA, NA, 0, NA))
##
## Coefficients:
##       ar1     ar2     ma1     ma2    sma1  sma2  intercept
##         0  0.5427  0.4914  0.2422  0.8514     0     0.0074
## s.e.    0  0.0692  0.0540  0.0739  0.0346     0     0.1145
##
## sigma^2 estimated as 0.1154:  log likelihood = -145.98,  aic = 301.97

Box.test(sarima2$residuals)

##
##  Box-Pierce test
##
```

```
## data:  sarima2$residuals
## X-squared = 0.00056788, df = 1, p-value = 0.981
```

The above series is simulated for the same number of points as of the original series (534) and since the 20% of the dataset is given to validation set, we will calculate the RMSE of the remaining dataset i.e. the last 107 data points.

```
library(CombMSC)

##
## Attaching package: 'CombMSC'

## The following object is masked from 'package:stats':
##
##     BIC

library(MLmetrics)

##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall

#Fitting the multiplicative sesonal 'SARIMA' model

mod1=sarima.Sim(n=534,period=12,
model=list(ar=c(0,0.5427),ma=c(0.4914,0.2422)),seasonal=list(ma=c(0.8514,0)))

tvec2=1:534

# Additive Seasonal component
mod2=(1.23435*sin(2*pi*tvec2/12))-(0.79405*cos(2*pi*tvec2/12))+ 5.70829

#Lambda from the Box-Cox transformation

mod =(mod1 + mod2)^(-.027)

#Validation set from the predicted
valid_mod = mod[428:534]

#Validation set from the true
valid_x = as.numeric(xk_true[428:534])


rmse = RMSE(valid_mod, valid_x)

rmse

## [1] 770.3787
```

The above is the value of RMSE on the validation set.

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

Department of Chemical Engineering

CH5350 Applied Time-Series Analysis (Jul – Nov 2018)

Honor Code

I SHRITEJ CHAVAN with Roll No. BE14B004, have worked out the project in all honesty without consulting any other person or referring to other resources for solutions.

Date: 03/12/2018

Signature of the student